

DEVICE AND METHOD FOR GENERATING EXPRESSION DATA IN OPERATION OF FINITE FIELD

Publication number: JP2000293507 (A)

Publication date: 2000-10-20

Inventor(s): KOGURE ATSUSHI

Applicant(s): FUJITSU LTD

Classification:


- **international:** *G09C1/00; G06F7/72; G06F11/10; G06F17/10; H03M13/01; G09C1/00; G06F7/60; G06F11/10; G06F17/10; H03M13/00; (IPC1-7): G06F17/10; G06F7/72; G06F11/10; G09C1/00; H03M13/01*


- **European:** G06F7/72F


Application number: JP19990102920 19990409

Priority number(s): JP19990102920 19990409


Also published as:

 JP3833412 (B2)

 EP1043654 (A2)

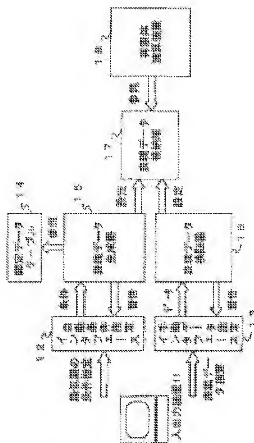
 EP1043654 (A3)

 US7142668 (B1)

 CA2298995 (A1)

Abstract of JP 2000293507 (A)

PROBLEM TO BE SOLVED: To flexibly set up the expression data of a finite field while reducing user's load as less as possible. **SOLUTION:** When a user inputs a specification condition from an I/O device 11 through an automatic condition specification interface 12, an expression data generator 15 refers to a fixed data table 14, and when there is no expression data satisfying the condition in the table 14, automatically generates expression data. The generated expression data are set up in an expression data storing area 17 and a finite field arithmetic unit 18 refers to the expression data and computes a finite field.



Data supplied from the **esp@cenet** database — Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-293507

(P2000-293507A)

(43) 公開日 平成12年10月20日 (2000.10.20)

(51) Int.Cl. ⁷	識別記号	F I	データベース ⁸ (参考)
G 0 6 F 17/10		G 0 6 F 15/31	Z 5 B 0 0 1
		7/72	5 B 0 5 6
	11/10	11/10	3 3 0 Q 5 J 0 6 5
G 0 9 C 1/00	6 5 0	G 0 9 C 1/00	6 5 0 A 5 J 1 0 4
H 0 3 M 13/01		H 0 3 M 13/01	9 A 0 0 1
		審査請求 未請求 請求項の数 9	O L (全 13 頁)

(21) 出願番号 特願平11-102920

(22) 出願日 平成11年4月9日 (1999.4.9)

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番1号

(72) 発明者 小暮 淳

神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

(74) 代理人 100074099

弁理士 大曾 義之 (外1名)

Fターム(参考) 5B001 AC01

5B056 AA00 AA04 BB01 BB74 HH00

5J065 AC01

5J104 AA22 NA08 NA18 NA27

9A001 EZ03 GG17

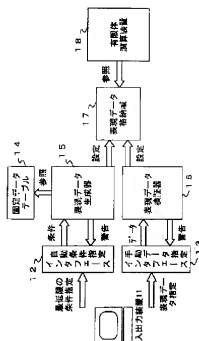
(54) 【発明の名称】 有限体演算における表現データ生成装置および方法

(57) 【要約】

【課題】 ユーザの負荷をなるべく軽減しながら、有限体の表現データを柔軟に設定することが課題である。

【解決手段】 ユーザが入出力装置11から自動条件指定インタフェース12を介して指定条件を入力すると、表現データ生成器15は、固定データテーブル14を参照し、固定データテーブル14に条件を満たす表現データがなければ、自動的に表現データを生成する。生成された表現データは表現データ格納域17に設定され、有限体演算装置18は、その表現データを参照して、有限体演算を行う。

第1のデータ生成装置の構成図



【特許請求の範囲】

【請求項1】 有限体を指定する条件を入力する入力手段と、

入力された条件に基づいて、前記有限体の表現データを自動的に生成する生成手段と、

生成された表現データを格納する表現データ格納手段とを備えることを特徴とするデータ生成装置。

【請求項2】 前記表現データ格納手段に格納された表現データに基づいて有限体演算を行う演算手段をさらに備えることを特徴とする請求項1記載のデータ生成装置。

【請求項3】 前記生成手段は、前記条件として前記有限体を記述する素数のビット長が入力されたとき、該ビット長に対応する素数データを自動的に生成し、前記表現データ格納手段に格納することを特徴とする請求項1記載のデータ生成装置。

【請求項4】 前記生成手段は、前記条件として前記有限体を記述する拡大次数が入力されたとき、該拡大次数に対応する既約多項式データを自動的に生成し、前記表現データ格納手段に格納することを特徴とする請求項1記載のデータ生成装置。

【請求項5】 前記生成手段は、最適化正規基底を使用する指示が入力されたとき、前記拡大次数に対応する最適化正規基底用既約多項式データを自動的に生成し、前記表現データ格納手段に格納することを特徴とする請求項4記載のデータ生成装置。

【請求項6】 あらかじめ決められた1つ以上の有限体の表現データを格納する固定データ格納手段をさらに備え、前記生成手段は、該固定データ格納手段に前記条件に対応する有限体の表現データが存在するとき、該条件に対応する有限体の表現データを前記表現データ格納手段に格納し、該固定データ格納手段に前記条件に対応する有限体の表現データが存在しないとき、該条件に対応する有限体の表現データを自動的に生成することを特徴とする請求項1記載のデータ生成装置。

【請求項7】 有限体の表現データを指定する指定手段と、指定された表現データが適切か否かを検証する検証手段とをさらに備え、該検証手段は、該指定された表現データが適切であれば、該指定された表現データを前記表現データ格納手段に格納し、該指定された表現データが適切でなければ、該指定手段に他の表現データを要求することを特徴とする請求項1記載のデータ生成装置。

【請求項8】 コンピュータのためのプログラムを記録した記録媒体であって、

有限体を指定する条件が入力されたとき、入力された条件に基づいて、該有限体の表現データを自動的に生成するステップと、生成された表現データを出力するステップとを含む処理を前記コンピュータに実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項9】 有限体を指定する条件を指定し、

指定された条件に基づいて、前記有限体の表現データを自動的に生成し、

生成された表現データを有限体演算装置に供給することを特徴とするデータ生成方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、符号、暗号等に関する有限体演算を行う際に、有限体の表現データを自動的に設定するデータ生成装置およびその方法に関する。

【0002】

【従来の技術】近年のコンピュータ技術およびネットワーク技術の発展に伴い、符号、暗号等を含む様々な分野において、有限体演算を行う局面が増えてきている。有限体 (finite field) とは、四則演算が定義されている有限集合を指し、有限体演算とは、有限体上で定義されたこれらの演算を指す。

【0003】特に、電子取引等のインターネットを利用したビジネスにおいては、オープンネットワークのセキュリティを確保するための暗号理論が非常に重要な技術として脚光を浴びている。暗号技術を実現するためには有限体演算が必須であるが、データのセキュリティを保つために非常に大きなサイズの有限体を用いるケースが多く、その演算効率を上げることは、実用上重要なテーマである。

【0004】有限体演算を実現するためには、まず、いくつかの表現データを決定して、有限体を表現しなければならない。有限体の要素 (元) の個数は、 p を素数とし、 m を正整数として、 p^m となる。この正整数 m は拡大次数と呼ばれる。一般に、有限体の要素を記述するためには、素数 p 、拡大次数 m 、および m 次の既約多項式 $F(x)$ を表現データとして設定しなければならないことが知られている。

【0005】従来の有限体演算装置においては、装置自身があらかじめ内蔵している固定の表現データを用いるか、もしくは、ユーザがすべての表現データを指定することで、有限体を表現していた。

【0006】

【発明が解決しようとする課題】しかしながら、従来の有限体の表現方法には、次のような問題がある。固定の表現データのみを用いる場合、符号、暗号等のシステムにおける様々な局面に必要となる多様な表現データを得ることができず、表現の柔軟性に欠ける。また、非常に大きな有限体において、ユーザがすべての表現データを指定する場合、表現データの選択に多大な労力を要する。さらに、ユーザが不適切な表現データを選択すると、演算速度が低下し、システムの性能が低下するという弊害が生じることもある。

【0007】本発明の課題は、効率の良い有限体演算を実現するために、ユーザの負担をなるべく軽減しなが

ら、有限体の表現データを柔軟に設定するデータ生成装置およびその方法を提供することである。

【0008】

【課題を解決するための手段】図1は、本発明のデータ生成装置の原理図である。図1のデータ生成装置は、入力手段1、生成手段2、および表現データ格納手段3を備える。入力手段1は、有限体を指定する条件を入力する。生成手段2は、入力された条件に基づいて、有限体の表現データを自動的に生成し、表現データ格納手段3は、生成された表現データを格納する。

【0009】生成手段2は、入力手段1が入力した指定条件に基づいて、その条件に対応する有限体の表現データを自動的に生成し、生成された表現データを表現データ格納手段3に格納する。格納された表現データは、有限体演算を行う装置に供給され、その表現データを用いた有限体演算が行われる。

【0010】ユーザは、入力手段1を介して、希望する有限体の条件を入力することができる。このとき、例えば、素数 p のビット長や拡大次数 m のような、直観的に思いつくことのできる最小限の条件さえ指定すれば、生成手段2が、素数 p や m 次の既約多項式 $F(x)$ のような、直観的に思いつくことの困難な表現データを自動的に生成する。

【0011】したがって、有限体の表現データの選択におけるユーザの負担が軽減され、生成手段2が生成可能な範囲内で、表現データを柔軟に設定することが可能になる。さらに、適切な条件が指定されれば、効率の良い有限体演算を実現する表現データが生成される。このように、本発明のポイントは、指定された条件に基づいて、有限体の表現データを自動的に生成することである。

【0012】例えば、図1の入力手段1は、後述する図4の入出力装置11および自動条件指定インタフェース

$$s = (g^7)^x = 9^4 = 5 \pmod{p}$$

また、ユーザBは、ユーザAから受信した $g^x \bmod p$

$$s = (g^7)^x = 5^6 = 5 \pmod{p}$$

ここでは、簡単のために $p=11$ としたが、実際には、 p として非常に大きな値が用いられる。 p が十分に大きければ、たとえ第3者が $g^x \bmod p$ と $g^y \bmod p$ を取得したとしても、これらの情報から x と y を求めることは極めて困難となり、秘密情報 s を知ることは事実上不可能である。こうして、ユーザAとユーザBは、秘密情報 s を安全に共有することができ、この情報を秘密鍵として用いて暗号通信を行うことができる。

【0018】次に、図3は、Diffie-Hellmanの鍵交換において、有限体 $GF(2^3) = \{0, 1\}$ の要素(0または1)を係数とする3次既約多項式を用いて表される有限体 $GF(2^3)$ を用いた例を示している。この場合、 $GF(2^3)$ は、 $GF(2)$ の要素を係数とする項からなる 2^3 個の2次以下の多項式を要素として持つ。

12に対応し、図1の生成手段2は、図4の表現データ生成器15に対応し、図1の表現データ格納手段3は、図4の表現データ格納域17に対応する。

【0013】

【発明の実施の形態】以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。まず、図2および図3を参照しながら、暗号技術に用いられる有限体演算の例を説明する。

【0014】図2は、よく知られたDiffie-Hellmanの鍵交換の例を示している。例えば、ユーザAとユーザBの間で秘密鍵暗号を用いてデータ通信を行う場合、まず、両者が秘密鍵を共有する必要がある。このとき、第三者に知られることなく、ネットワーク上で秘密情報を交換するために、Diffie-Hellmanの鍵交換が用いられる。

【0015】素数 p 、拡大次数 m により表される要素数 p^m の有限体を $GF(p^m)$ と記すことにすると、図2で用いられている有限体は、 $GF(11)$ と記される($m=1$)。 $m=1$ の有限体は、素体(prime field)とも呼ばれる。

【0016】ここでは、 $p=11$ 、 $g=2$ が公開されており、ユーザAの秘密情報が $x=4$ であり、ユーザBの秘密情報が $y=6$ であり、共有すべき秘密情報が $s = g^{xy} \bmod p = 2^{24} \bmod 11 = 5$ である場合を考える。 $\bmod p$ は、 p を法とする剰余演算を表し、 $g^{xy} \bmod p$ は、 $GF(11)$ における有限体演算の一例を表している。

【0017】このとき、ユーザAは、 $g^x \bmod p = 2^4 \bmod 11 = 5$ をユーザBに送信し、ユーザBは、 $g^y \bmod p = 2^6 \bmod 11 = 9$ をユーザAに送信する。次に、ユーザAは、ユーザBから受信した $g^y \bmod p = 9$ を用いて、次式により共有秘密情報 s を求める。

$$p = 5 \text{ を用いて、次式により共有秘密情報 } s \text{ を求める。}$$

$$p = 5 \text{ (2)}$$

【0019】ここでは、 $p=2$ 、3次既約多項式 $f(x) = x^3 + x + 1$ 、 $g = x \bmod f(x)$ が公開されており、ユーザAの秘密情報が $i=4$ であり、ユーザBの秘密情報が $j=6$ であり、共有すべき秘密情報が $s = g^{ij} \bmod f(x)$ である場合を考える。 $\bmod f(x)$ は、 $f(x)$ を法とする多項式の剰余演算を表し、 $g^{ij} \bmod f(x)$ は、 $GF(2^3)$ における有限体演算の一例を表している。

【0020】このとき、ユーザAは、 $g^i \bmod f(x) = x^4 \bmod (x^3 + x + 1) = x^2 + x$ をユーザBに送信し、ユーザBは、 $g^j \bmod f(x) = x^6 \bmod (x^3 + x + 1) = x^2 + 1$ をユーザAに送信する。次に、ユーザAは、ユーザBから受信した $g^j \bmod f(x) = x^2 + 1$ を用いて、次式により共

有秘密情報sを求める。

$$s = (g^i)^j = (x^2 + 1)^4 = x + 1 \pmod{f(x)} \quad (3)$$

また、ユーザBは、ユーザAから受信した $g^i \bmod f(x) = x^2 + x$ を用いて、次式により共有秘密情報

$$s = (g^j)^i = (x^2 + x)^5 = x + 1 \pmod{f(x)} \quad (4)$$

ここでは、簡単のために $m=3$ としたが、実際には、 m として非常に大きな値が用いられる。 m が十分に大きければ、たとえ第3者が $g^i \bmod f(x)$ と $g^j \bmod f(x)$ を取得したとしても、これらの情報から i と j を求めることは極めて困難となり、第3者が秘密情報sを知ることは事実上不可能である。こうして、ユーザAとユーザBは、秘密情報sを安全に共有することができ、この情報を秘密鍵として用いて暗号通信を行うことができる。

【0021】上述したように、素数GF(p)を表現するためには、素数pを指定しなければならず、有限体GF(2^m)を表現するためには、拡大次数mとm次の既約多項式を指定しなければならない。しかしながら、pやmとして非常に大きな数を用いる場合、ユーザが素数pやm次の既約多項式を隅に指定することは難しい。

【0022】本実施形態のデータ生成装置は、ユーザが直観的に思いつくことのできる最小限の条件さえ指定すれば、自動的に素数pやm次の既約多項式等の表現データを生成する。このため、ユーザが表現データを指定する際の負担が大きく軽減される。

【0023】このとき、可能であれば、高速演算に適した表現データを選択することで、より高速な有限体演算が実現される。さらに、ユーザがすべての表現データを指定することのできるインタフェースを設けることにより、様々な局面に対応できる柔軟性が実現される。

【0024】図4は、このようなデータ生成装置の基本構成図である。図4のデータ生成装置は、入出力装置11、自動条件指定インタフェース12、手動データ指定インタフェース13、固定データテーブル14、表現データ生成器15、表現データ検証器16、および表現データ格納域17を備え、有限体演算装置18が用いる有限体の表現データを生成する。

【0025】ユーザは、入出力装置11を介して、自動条件指定インタフェース12または手動データ指定インタフェース13に対して必要な指示を入力する。自動条件指定インタフェース12は、ユーザが表現データ生成のための最小限の条件を指定する場合に用いられ、手動データ指定インタフェース13は、ユーザが表現データの任意のパラメータを自ら指定する場合に用いられる。

【0026】固定データテーブル14は、あらかじめ決められた適切な表現データを格納し、表現データ生成器15は、自動条件指定インタフェース12による指定に従って、最終的に使用される表現データを決定する。表現データ検証器16は、手動データ指定インタフェース13により指定された表現データが有限体演算に適切な

ものかどうかを検証する。

【0027】表現データ格納域17は、実際の演算時に使用される表現データを格納し、有限体演算装置18は、表現データ格納域17の表現データを参照して、有限体演算を行う。

【0028】図5は、図4のデータ生成装置による処理のフローチャートである。ユーザは表現データ選択のための条件指定において、自動モードまたは手動モードのいずれかを選択することができ、データ生成装置は、自動条件指定インタフェース12および手動データ指定インタフェース13により、どちらのモードが選択されたかをチェックする(ステップS1)。自動モードにおいては、ユーザは最低限の条件を指定するだけで済み、手動モードにおいては、ユーザは表現データのすべてのパラメータを指定しなければならない。

【0029】自動モードが選択された場合、表現データ生成器15は、まず、指定された条件が適切か否か(正當か否か)をチェックする(ステップS2)。そして、それが適切でなければ、自動条件指定インタフェース12を介して警告を発し、条件の再入力要求(ステップS3)、ステップS2以降の処理を繰り返す。

【0030】適切な条件が入力されると、次に、固定データテーブル14を検索して、指定された条件を満たす表現データが存在するか否かをチェックする(ステップS4)。そのような表現データが固定データテーブル14に存在すれば、そのパラメータを表現データ格納域17に設定し(ステップS6)、処理を終了する。また、そのような表現データが存在しなければ、指定された条件に基づいて表現データを生成して(ステップS5)、表現データ格納域17に設定し(ステップS6)、処理を終了する。

【0031】固定データテーブル14にあらかじめ保持された表現データは、例えば、高速な有限体演算を実現できるように選ばれたものであり、表現データ生成器15による表現データ生成のロジックは、例えば、高速な有限体演算を実現する表現データを生成するものである。

【0032】ステップS1において手動モードが選択された場合、表現データ検証器16は、指定された表現データの正当性を検証し(ステップS7)、それが有限体演算を行うにあたって適切なものかどうかをチェックする(ステップS8)。

【0033】そして、その表現データが適切であれば、それを表現データ格納域17に設定し(ステップS6)、処理を終了する。また、指定された表現データが

適切でなければ、手動データ指定インタフェース13を介して警告を発し、再度、適切な表現データを指定するようにユーザに促し（ステップS9）、ステップS7以降の処理を繰り返す。

【0034】表現データが表現データ格納域17に設定されると、有限体演算部18は、その値に基づいて有限体演算を行う。自動モードが選択された場合、自動的に高速な有限体演算を実現する表現データが設定されるため、演算の効率が向上する。

【0035】次に、図6から図12までを参照しながら、図4の構成に基づく様々なデータ生成装置の実施形態について説明する。図6は、素数 p に関する条件に基づいて素体GF(p)の表現データを生成するデータ生成装置の構成図である。素体GF(p)の表現データとしては、素数 p を設定すればよい。この数 p は、標数(characteristic)とも呼ばれる。

【0036】図6のデータ生成装置は、入力装置11、自動条件指定インタフェース12、手動データ指定インタフェース13、素数表21、テーブル参照部22、乱数発生部23、素数判定部24、25、および素数格納域26を備え、有限体演算装置18が用いる素体の表現データを生成する。

【0037】素数表21は、図4の固定データテーブル14に対応し、あらかじめ決められた素数とそのバイナリコードのビット長 n の対応関係を格納する。例えば、図7に示す素数表においては、 $n=2, 3, 4, \dots$ に対応する素数3, 7, 11, \dots が順に格納されている。

【0038】テーブル参照部22、乱数発生部23、および素数判定部24は、図4の表現データ生成器15に対応する。テーブル参照部22は、与えられたビット長 n をキーとして素数表21を参照し、対応する素数取得する。与えられたビット長 n に対応する素数が素数表21に存在しないとき、乱数発生部23は、ビット長 n の乱数を生じ、素数判定部24は、その乱数が素数かどうかを判定する。

【0039】素数判定部25は、図4の表現データ検証器16に対応し、ユーザが指定した素数 p が本当に素数であるかどうかを判定する。素数格納域26は、図4の表現データ格納域17に対応し、与えられた素数 p を表現データとして格納する。

【0040】図8は、図6のデータ生成装置による処理のフローチャートである。まず、データ生成装置は、自動モードと手動モードのいずれが選択されたかをチェックする(ステップS11)。自動モードにおいては、ユーザは自動条件指定インタフェース12を介して素数のビット長 n (素体のビットサイズ)のみを入力し、手動モードにおいては、ユーザは自ら具体的な素数 p を選択し、手動データ指定インタフェース13を介してそれを入力する。

【0041】自動モードが選択された場合、テーブル参照部22は、指定されたビット長 n がサポート範囲内のビット長かどうかをチェックする(ステップS12)。そして、それがサポート範囲外であれば、自動条件指定インタフェース12を介して警告を発し、ビット長の再入力を要求して(ステップS13)、ステップS12以降の処理を繰り返す。

【0042】サポート範囲内のビット長 n が入力されると、次に、素数表21を検索して、ビット長 n に対応する素数が存在するかどうかをチェックする(ステップS14)。そのような素数が素数表21に存在すれば、それを素数 p として素数格納域26に設定し(ステップS16)、処理を終了する。

【0043】また、そのような素数が存在しなければ、乱数発生部23は、指定されたビット長 n の乱数を生じさせ、素数判定部24は、発生した乱数が素数かどうかを判定する(ステップS15)。そして、素数判定部24は、素数と判定された乱数を表現データとして採用し、それを素数 p として素数格納域26に設定して(ステップS16)、処理を終了する。

【0044】素数判定部24による素数判定(primality test)のアルゴリズムとしては、例えば、以下の文献に記載されているようなものが用いられる。

文献[1]:情報処理学会 監修、岡本龍明(おかもと たつあき)・太田和夫(おた かずお) 共編、"暗号・ゼロ知識証明・数論", 共立出版, p. 1, 30-143, 1995.

文献[2]:IEEE P1363 Annex A/Editorial Contribution, "Standard Specifications For Public Key Cryptography", pp. 78-81, 1998.

ステップS11において手動モードが選択された場合、素数判定部25は、素数判定部24と同様に、指定された数 p の素数判定を行い(ステップS17)、素数が本当に素数であるかどうかをチェックする(ステップS18)。

【0045】そして、指定された数 p が素数であれば、それを表現データとして採用して、素数格納域26に設定し(ステップS16)、処理を終了する。また、それが素数でなければ、手動データ指定インタフェース13を介して警告を発し、再度、正しい素数を入力するようにユーザに促し(ステップS19)、ステップS17以降の処理を繰り返す。

【0046】素数 p が素数格納域26に設定されると、有限体演算部18は、その値に基づいて素体GF(p)上の有限体演算を行う。このような構成によれば、ユーザがビット長さ n を指定すれば自動的に素数 p が生成されるため、ユーザが素数 p を指定する際の負荷が軽減される。また、必要であれば、ユーザが素数 p を直接指定することもでき、柔軟な操作性が実現される。

【0047】図9は、拡大次数 m に基づいて有限体GF

(2^n) の表現データを生成するデータ生成装置の構成図である。有限体 GF (2^n) の場合、素数 p は 2 に決められているため、それ以外の表現データとしては、拡大次数 m と m 次の既約多項式 $F(x)$ を設定すればよい。 m 次の多項式は、例えば、各項 x^k ($k=0, 1, \dots, m$) の係数 (0 または 1) の集合により表すことができる。

【0048】図 9 のデータ生成装置は、入出力装置 11、自動条件指定インタフェース 12、手動データ指定インタフェース 13、既約多項式表 31、テーブル参照部 32、多項式発生部 33、既約性判定部 34、35、および既約多項式格納域 36 を備え、有限体演算装置 18 が用いる有限体の表現データを生成する。

【0049】既約多項式表 31 は、図 4 の固定データテーブル 14 に対応し、あらかじめ決められた拡大次数 m と m 次の既約多項式 $F(x)$ の対応関係を格納する。例えば、図 10 に示す既約多項式表においては、 $m=2$ 、3、4、... に対応する既約多項式として、 x^2+x+1 、 x^3+x+1 、 x^4+x+1 、... が格納されている。

【0050】テーブル参照部 32、多項式発生部 33、および既約性判定部 34 は、図 4 の表現データ生成器 15 に対応する。テーブル参照部 32 は、与えられた拡大次数 m をキーとして既約多項式表 31 を参照し、対応する既約多項式を取得する。与えられた拡大次数 m に対応する既約多項式が既約多項式表 31 に存在しないとき、多項式発生部 33 は、 m 次の多項式を発生し、既約性判定部 34 は、その多項式が既約か否かを判定する。

【0051】既約性判定部 35 は、図 4 の表現データ検証器 16 に対応し、ユーザが指定した m 次の既約多項式が本当に既約であるか否かを判定する。既約多項式格納域 36 は、図 4 の表現データ格納域 17 に対応し、拡大次数 m と m 次の既約多項式 $F(x)$ を表現データとして格納する。

【0052】図 11 は、図 9 のデータ生成装置による処理のフローチャートである。まず、データ生成装置は、自動モードと手動モードのいずれが選択されたかをチェックする (ステップ S21)。自動モードにおいては、ユーザは自動条件指定インタフェース 12 を介して拡大次数 m のみを入力し、手動モードにおいては、ユーザは自ら具体的な m 次の既約多項式 $F(x)$ を選択し、それを拡大次数 m とともに手動データ指定インタフェース 13 を介して入力する。

【0053】自動モードが選択された場合、テーブル参照部 32 は、指定された拡大次数 m がサポート範囲内の拡大次数か否かをチェックする (ステップ S22)。そして、それがサポート範囲外であれば、自動条件指定インタフェース 12 を介して警告を発し、拡大次数の再入力を要求して (ステップ S23)、ステップ S22 以降の処理を繰り返す。

【0054】サポート範囲内の拡大次数 m が入力されると、次に、既約多項式表 31 を検索して、拡大次数 m に対応する既約多項式が存在するか否かをチェックする (ステップ S24)。そのような既約多項式が既約多項式表 31 に存在すれば、それを既約多項式 $F(x)$ として、拡大次数 m とともに既約多項式格納域 36 に設定し (ステップ S26)、処理を終了する。

【0055】また、そのような既約多項式が存在しなければ、多項式発生部 33 は、 m 次の多項式を発生させ、既約性判定部 34 は、発生した多項式が既約か否かを判定する (ステップ S25)。そして、既約性判定部 34 は、既約と判定された多項式を既約多項式 $F(x)$ として採用し、それを拡大次数 m とともに既約多項式格納域 36 に設定して (ステップ S26)、処理を終了する。

【0056】ステップ S21 において手動モードが選択された場合、既約性判定部 35 は、既約性判定部 34 と同様にして、指定された多項式 $F(x)$ の既約性判定 (irreducibility test) を行い (ステップ S27)、それが本当に既約であるか否かをチェックする (ステップ S28)。

【0057】そして、指定された多項式 $F(x)$ が既約であれば、それを表現データとして採用して、拡大次数 m とともに既約多項式格納域 36 に設定し (ステップ S26)、処理を終了する。また、それが既約でなければ、手動データ指定インタフェース 13 を介して警告を発し、再度、正しい既約多項式を入力するようにユーザに促し (ステップ S29)、ステップ S27 以降の処理を繰り返す。

【0058】図 12 は、図 11 のステップ S25 における既約多項式生成処理のフローチャートである。まず、テーブル参照部 32 は、多項式発生部 33 に拡大次数 m を入力し (ステップ S31)、多項式発生部 33 は、 $(m+1)$ ビットの乱数 (バイナリコード) を発生させる (ステップ S32)。

【0059】次に、多項式発生部 33 は、有限体 GF (2) = {0, 1} の要素を係数とする m 次の多項式を生成する (ステップ S33)。ここでは、 $k=1, 2, \dots, m+1$ のそれぞれの値について、発生した乱数の k ビット目の値 (0 または 1) を x^{k-1} の項の係数とし、それらの項からなる m 次の多項式を生成する。

【0060】次に、既約性判定部 34 は、生成された多項式の既約性判定を行い (ステップ S34)、それが既約か否かをチェックする (ステップ S35)。既約性判定のアルゴリズムとしては、例えば、上述した文献「2」の 30 ページに記載されているようなものが用いられる。生成された多項式が既約でなければ、ステップ S32 以降の処理を繰り返し、既約な多項式が得られれば、それを既約多項式格納域 36 に出力して (ステップ S36)、処理を終了する。

【0061】図11の処理により、拡大次数 m と既約多項式 $F(x)$ が既約多項式格納域36に設定されると、有限体演算部18は、そのデータに基づいて有限体 $GF(2^m)$ 上の有限体演算を行う。このような構成によれば、ユーザが拡大次数 m を指定すれば自動的に m 次の既約多項式 $F(x)$ が生成されるため、ユーザが既約多項式 $F(x)$ を指定する際の負荷が軽減される。また、必要であれば、ユーザが既約多項式 $F(x)$ を直接指定することもでき、柔軟な操作性が実現される。

【0062】ところで、有限体 $GF(2^m)$ 上の演算を高速化する方法の1つとして、最適化正規基底(optimal normal basis)を用いる方法が知られている(文献[1], p.p. 167-170/文献[2], p.p. 19-24)。一般に、有限体 $GF(2^m)$ は、正規基底(normal basis)を生成元として表現することができ、特に、最適化正規基底に基づく表現データを用いれば、必要乗算および乗算を飛躍的に高速化することができる。

【0063】そこで、図9のデータ生成装置において、最適化正規基底を用いた表現データを生成するオプションを設けることにする。ただし、最適化正規基底は、必ずしもすべての拡大次数について存在するわけではなく、これを利用するためには、拡大次数 m を適切に指定する必要がある。例えば、略号に用いられる $160 \leq m \leq 2000$ の拡大次数のうち最適化正規基底が存在するものは、以下の文献にリストアップされている。

文献[3]: American National Standard, x9.62-199x, Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm, p.41, Working Draft.

ユーザは、自動条件指定インタフェース12および手動データ指定インタフェース13を介して、最適化正規基底を使用するか否かをオプションとして指定することができる。この場合、既約多項式表31には、各拡大次数について、最適化正規基底が存在するか否かを表す情報が付加され、存在する場合は、必要に応じてその最適化正規基底用既約多項式の情報が付加される。最適化正規基底用既約多項式とは、最適化正規基底を用いて有限体を表現するために必要な特定の既約多項式を指す。

【0064】図13は、このようなデータ生成装置による処理のフローチャートである。まず、データ生成装置は、自動モードと手動モードのいずれが選択されたかをチェックする(ステップS41)。自動モードにおいては、ユーザは、自動条件指定インタフェース12を介して、拡大次数 m と最適化正規基底を使用する指示とを入力する。また、手動モードにおいては、ユーザは、自ら具体的な m 次の最適化正規基底用既約多項式 $F(x)$ を選択し、それを拡大次数 m とともに手動データ指定インタフェース13を介して入力する。

【0065】自動モードが選択された場合、テーブル参

照部32は、既約多項式表31を参照して、指定された拡大次数 m の最適化正規基底が存在するか否かをチェックする(ステップS42)。そして、 m 次の最適化正規基底が存在しないか、あるいは拡大次数 m がサポート範囲外であれば、自動条件指定インタフェース12を介して警告を発し、拡大次数の再入力要求して(ステップS43)、ステップS42以降の処理を繰り返す。

【0066】最適化正規基底を有する拡大次数 m が入力されると、次に、既約多項式表31に最適化正規基底用既約多項式が存在するか否かをチェックする(ステップS44)。そのような既約多項式が既約多項式表31に存在すれば、それを最適化正規基底用既約多項式 $F(x)$ として、拡大次数 m とともに既約多項式格納域36に設定し(ステップS46)、処理を終了する。

【0067】また、そのような既約多項式が存在しなければ、多項式発生部33は、 m 次の最適化正規基底用既約多項式を生成する(ステップS45)。そして、それを最適化正規基底用既約多項式 $F(x)$ として、拡大次数 m とともに既約多項式格納域36に設定し(ステップS46)、処理を終了する。最適化正規基底用既約多項式の生成アルゴリズムとしては、例えば、上述した文献[2]の37ページに記載されているようなものが用いられる。

【0068】ステップS41において手動モードが選択された場合、既約性判定部35は、指定された多項式 $F(x)$ の最適化正規基底用既約性判定を行い(ステップS47)、それが本当に最適化正規基底用既約多項式であるか否かをチェックする(ステップS48)。

【0069】最適化正規基底用既約性判定は、例えば、上述した文献[2]の20ページに記載されている最適化正規基底が存在するか否かの判定アルゴリズムと、最適化正規基底用既約多項式の生成アルゴリズムとを組み合わせて行われる。

【0070】この場合、既約性判定部35は、まず、指定された多項式 $F(x)$ の拡大次数 m において、最適化正規基底が存在するか否かをチェックする。最適化正規基底が存在すれば、次に、 m 次の最適化正規基底用既約多項式を生成して、 $F(x)$ を生成された多項式と比較する。そして、両者が一致すれば、 $F(x)$ が最適化正規基底用既約多項式であると判定する。

【0071】また、あらかじめこれらのアルゴリズムに基づいて、サポート範囲内のすべての最適化正規基底用既約多項式のテーブルを生成しておき、そのテーブルを参照しながら、最適化正規基底用既約性判定を行うこともできる。

【0072】指定された多項式 $F(x)$ が最適化正規基底用既約多項式であれば、それを表現データとして採用して、拡大次数 m とともに既約多項式格納域36に設定し(ステップS46)、処理を終了する。また、それが最適化正規基底用既約多項式でなければ、手動データ指

定インタフェース13を介して警告を発し、再度、正しい最適化正規基底用既約多項式を入力するようにユーザに促し(ステップS49)、ステップS47以降の処理を繰り返す。

【0073】拡大次数 m と最適化正規基底用既約多項式 $F(x)$ が既約多項式格納域36に設定されると、有限体演算部18は、そのデータに基づいて有限体 $GF(2^n)$ 上の有限体演算を行う。

【0074】このような構成によれば、ユーザが最適化正規基底を有する拡大次数 m さえ指定すれば、自動的に m 次の最適化正規基底用既約多項式 $F(x)$ が生成される。このため、ユーザが最適化正規基底用既約多項式 $F(x)$ を指定する際の負荷が軽減されるとともに、最適化正規基底に基づく高速な有限体演算が可能となる。また、必要であれば、ユーザが最適化正規基底用既約多項式 $F(x)$ を直接指定することもでき、柔軟な操作性が実現される。

【0075】以上の実施形態においては、主として、有限体 $GF(p)$ および $GF(2^n)$ の表現データを生成する場合について説明したが、一般の有限体 $GF(p^n)$ の表現データを生成する場合も同様である。この場合、ユーザは、素数 p またはそのビット長 n と拡大次数 m さえ指定すれば、データ生成装置は、自動的に $GF(p^n)$ の表現データを生成する。

【0076】ところで、本実施形態のデータ生成装置および有限体演算装置は、図14に示すような情報処理装置(コンピュータ)を用いて構成することができる。図14の情報処理装置は、CPU(中央処理装置)41、メモリ42、入力装置43、出力装置44、外部記憶装置45、媒体駆動装置46、およびネットワーク接続装置47を備え、それらはバス48により互いに接続されている。

【0077】メモリ42は、例えば、ROM(read only memory)、RAM(random access memory)等を含み、処理に用いられるプログラムとデータを格納する。CPU41は、メモリ42を利用してプログラムを実行することにより、必要な処理を行う。

【0078】この場合、図4の自動条件指定インタフェース12、手動データ指定インタフェース13、表現データ生成器15、および表現データ検証器16は、メモリ42の特定のプログラムコードセグメントに格納されたインストラクションの集合により実現されるソフトウェアコンポーネントに対応する。また、図4の固定データテーブル14と表現データ格納域17は、メモリ42内に設けられる。

【0079】入力装置43は、例えば、キーボード、ポインティングデバイス、タッチパネル等であり、ユーザからの指示や情報の入力に用いられる。出力装置44は、例えば、ディスプレイ、プリンタ、スピーカ等であり、ユーザへのメッセージや処理結果の出力に用いられ

る。

【0080】外部記憶装置45は、例えば、磁気ディスク装置、光ディスク装置、光磁気ディスク(magnetooptical disk)装置等である。情報処理装置は、この外部記憶装置45に、上述のプログラムとデータを保存しておき、必要に応じて、それらをメモリ42にロードして使用することができる。

【0081】媒体駆動装置46は、可搬記録媒体49を駆動し、その記録内容にアクセスする。可搬記録媒体49としては、メモリアカード、フロッピーディスク、CD-ROM(compact disk read only memory)、光ディスク、光磁気ディスク等、任意のコンピュータ読み取り可能な記録媒体が用いられる。ユーザは、この可搬記録媒体49に上述のプログラムとデータを格納しておき、必要に応じて、それらをメモリ42にロードして使用することができる。

【0082】ネットワーク接続装置47は、任意のネットワーク(回線)を介して外部の装置と通信し、通信に伴うデータ変換を行う。情報処理装置は、必要に応じて、ネットワーク接続装置47を介して上述のプログラムとデータを外部の装置から受け取り、それらをメモリ42にロードして使用することができる。

【0083】図15は、図14の情報処理装置にプログラムとデータを供給することのできるコンピュータ読み取り可能な記録媒体を示している。可搬記録媒体49や外部のデータベース50に保存されたプログラムとデータは、メモリ42にロードされる。そして、CPU41は、そのデータを用いてそのプログラムを実行し、必要な処理を行う。

【0084】

【発明の効果】本発明によれば、ユーザが最小限の条件さえ指定すれば自動的に有限体の表現データが生成されるため、ユーザの負荷が大きく軽減される。また、特定の表現データを選択的に生成することにより、高速な有限体演算を実現することができる。さらに、有限体演算の様々な局面に応じて、ユーザがすべての表現データを柔軟に指定することもできる。

【図面の簡単な説明】

【図1】本発明のデータ生成装置の原理図である。

【図2】第1の秘密情報の交換を示す図である。

【図3】第2の秘密情報の交換を示す図である。

【図4】第1のデータ生成装置の構成図である。

【図5】第1のデータ生成処理のフローチャートである。

【図6】第2のデータ生成装置の構成図である。

【図7】素数表を示す図である。

【図8】第2のデータ生成処理のフローチャートである。

【図9】第3のデータ生成装置の構成図である。

【図10】既約多項式表を示す図である。

【図11】第3のデータ生成処理のフローチャートである。

【図12】既約多項式生成処理のフローチャートである。

【図13】第4のデータ生成処理のフローチャートである。

【図14】情報処理装置の構成図である。

【図15】記録媒体を示す図である。

【符号の説明】

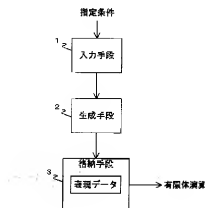
- 1 入力手段
- 2 生成手段
- 3 表現データ格納手段
- 11 入出力装置
- 12 自動条件指定インタフェース
- 13 手動データ指定インタフェース
- 14 固定データテーブル
- 15 表現データ生成器
- 16 表現データ検証器
- 17 表現データ格納域
- 18 有限体演算装置

- 21 素数表
- 22、32 テーブル参照部
- 23 乱数発生部
- 24、25 素数判定部
- 26 素数格納域
- 31 既約多項式表
- 33 多項式発生部
- 34、35 既約性判定部
- 36 既約多項式格納域
- 41 CPU
- 42 メモリ
- 43 入力装置
- 44 出力装置
- 45 外部記憶装置
- 46 媒体駆動装置
- 47 ネットワーク接続装置
- 48 バス
- 49 可搬記録媒体
- 50 データベース

//

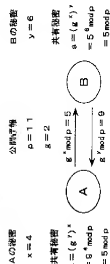
【図1】

本 発 明 の 原 理 図

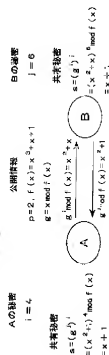


【図2】

第1の秘密情報の交換を示す図

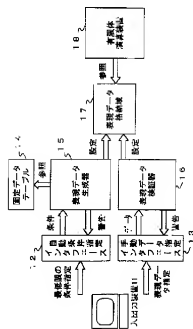


【図3】

第2の秘密情報の
交換を示す図

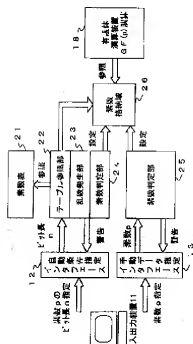
【図4】

第1のデータ生成装置の構成図



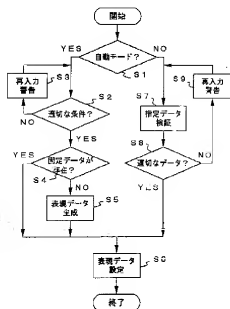
【図5】

第2のデータ生成装置の構成図



【図6】

第1のデータ生成処理のフローチャート



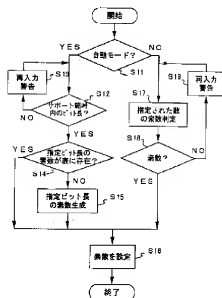
【図7】

素数表を示す図

ビット長n	nビット素数
2	3
3	7
4	11
⋮	⋮

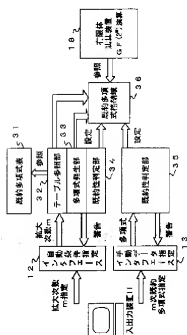
【図8】

第2のデータ生成処理のフローチャート



【図9】

第3のデータ生成装置の構成図



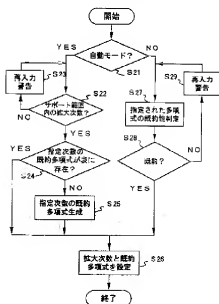
【図10】

既約多項式表を示す図

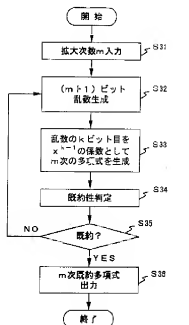
拡大次数n	m次既約多項式
2	$x^2 + x + 1$
3	$x^3 + x + 1$
4	$x^4 + x + 1$
⋮	⋮

【図11】

第3のデータ生成処理のフローチャート

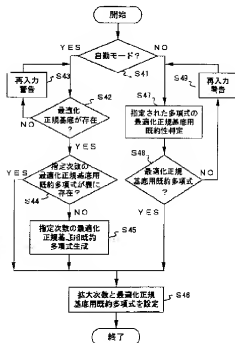


【図12】

既約多項式生成処理の
フローチャート

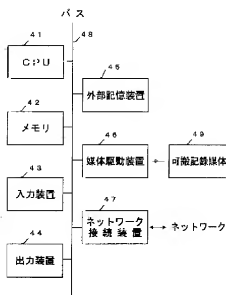
【図13】

第4のデータ生成処理のフローチャート



【図14】

情報処理装置の構成図



【図15】

記録媒体を示す図

